```
INSERT INTO RiIDF(token, idf)
SELECT   T.token, LOG(S.size)-LOG(COUNT(UNIQUE(*)))
FROM     RiTokens T, RiSize S
GROUP BY T.token, S.size
```

(a) Relation with token *idf* counts

```
INSERT INTO RiLength(tid, len)
SELECT   T.tid, SQRT(SUM(I.idf*I.idf*T.tf*T.tf))
FROM     RiIDF I, RiTF T
WHERE    I.token = T.token
GROUP BY T.tid
```

(c) Relation with weight-vector lengths

```
INSERT INTO RiSum(token, total)
SELECT   R.token, SUM(R.weight)
FROM     RiWeights R
GROUP BY R.token
```

(e) Relation with total token weights

```
INSERT INTO RiTF(tid, token, tf)
SELECT   T.tid, T.token, COUNT(*)
FROM     RiTokens T
GROUP BY T.tid, T.token
```

(b) Relation with token *tf* counts

```
INSERT INTO RiWeights(tid, token, weight)
SELECT   T.tid, T.token, I.idf*T.tf/L.len
FROM     RiIDF I, RiTF T, RiLength L
WHERE    I.token = T.token AND T.tid = L.tid
```

(d) Final relation with normalized tuple weight vectors

```
INSERT INTO RiSize(size)
SELECT   COUNT(*)
FROM     Ri
```

(f) Dummy elation used to create RiIDF

FIG. 1

```
SELECT    r1w.tid AS tid1, r2w.tid AS tid2
FROM      R1Weights r1w,R2Weights r2w
WHERE     r1w.token = r2w.token
GROUP BY  r1w.tid, r2w.tid
HAVING    SUM(r1w.weight*r2w.weight≥ Φ
```

FIG. 2

```
SELECT    rw.tid, rw.token, rw.weight/rs.total AS P
FROM      RiWeights rw, RiSum rs
WHERE     rw.token = rs.token
```

FIG. 3

```
INSERT INTO RiSample(tid,token,c)
 SELECT   rw.tid, rw.token, ROUND(S *  rw.weight/rs.total, 0) AS c
 FROM     RiWeights rw, RiSum rs
 WHERE    rw.token = rs.token
```

FIG. 4

```
SELECT  r1w.tid AS tidi,r2s.tid AS tid2
FROM    R1weights r1w, R2sample r2s, R2sum r2sum, R1V r1v
WHERE   r1w.token = r2s.token AND r1w.token = r2sum.token AND r1w.tid = r1v.tid
```

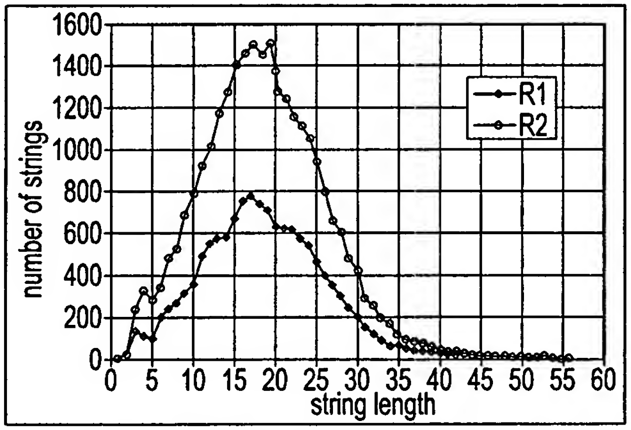
FIG. 5

```
SELECT tid1, tid2
FROM
(
  SELECT   r1w.tid AS tid1, r2s.tid AS tid2, SUM(r1w.weight * r2sum.total) AS Ci
  FROM     R1weights r1w, R2sample r2s, R2sum r2sum
  WHERE    r1w.token = r2s.token AND r1w.token = r2sum.token AND r1w.tid = r1v.tid
  GROUP BY r1w.tid, r2s.tid
UNION ALL
  SELECT   r1s.tid AS tid1, r2w.tid AS tid2, SUM(r2w.weight * r1sum.total) AS Ci
  FROM     R2weights r2w, R1sample r1s, R1sum r1sum
  WHERE    r2w.token = r1s.token AND r2w.token = r1sum.token AND r2w.tid = r2v.tid
  GROUP BY r2w.tid, r1s.tid
) SYM
GROUP BY tid1, tid2
HAVING AVG(Ci) ≥ S * Φ'
```

FIG. 6

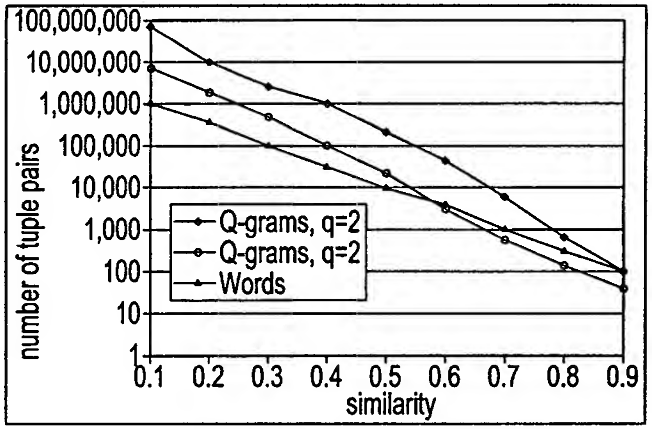
```
SELECT   r1s.tid AS tid1, r2s.tid AS tid2
FROM     R1Sample r1s, R2Sample r2s, R1Sum r1sum, R2Sum r2sum
WHERE    r1s.token = r1sum.token AND R2Sample.token = r2sum.token AND r1s.token = r2s.token
GROUP BY r1s.tid, r2s.tid
HAVING   SUM(r1sum.total * r2sum.total) ≥ S * S * Φ'
```

FIG. 7



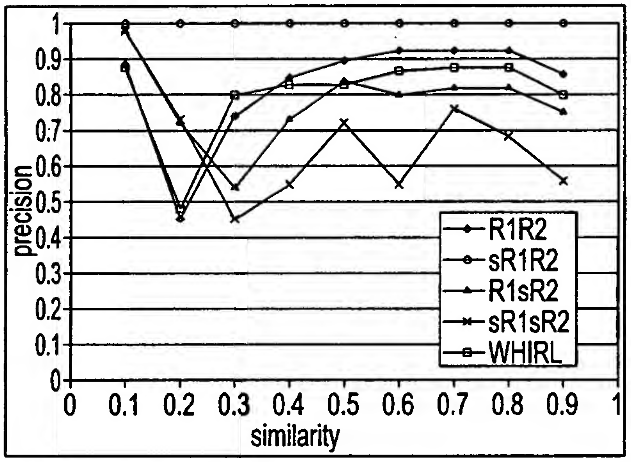
(a) String lengths in data sets R_1 and R_2.

FIG. 8A



The size of $R_1 x_\phi R_2$ for different similarity thresholds and token choices.

FIG. 8B



(a) Words

FIG. 9A

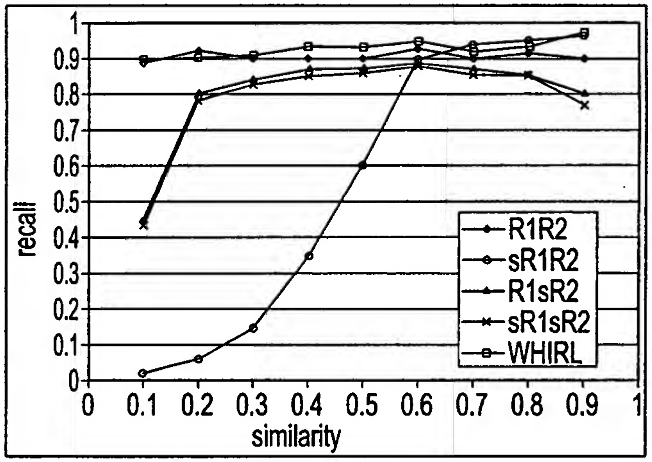


FIG. 9B

TEXT JOINS FOR DATA CLEANSING AND INTEGRATION IN A RELATIONAL DATABASE MANAGEMENT SYSTEM
10/828,819 (1209-29)

Replacement Sheet

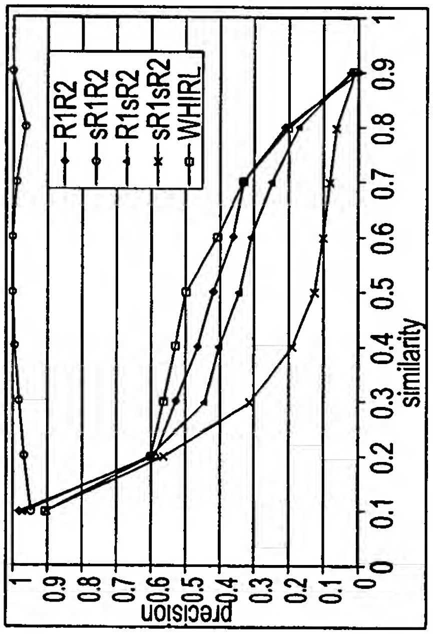


FIG. 9C

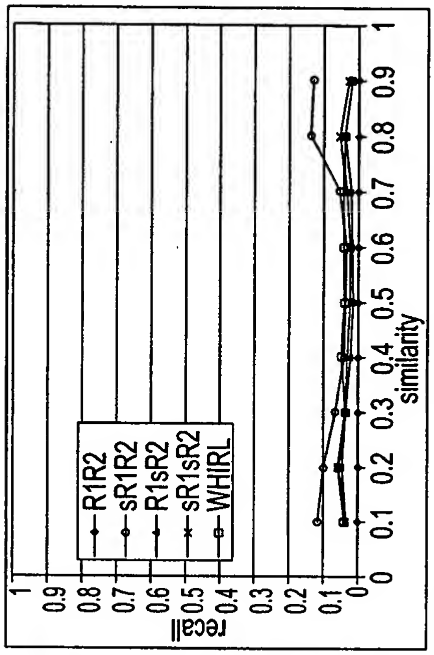


FIG. 9D

(b) Q-grams with q = 2

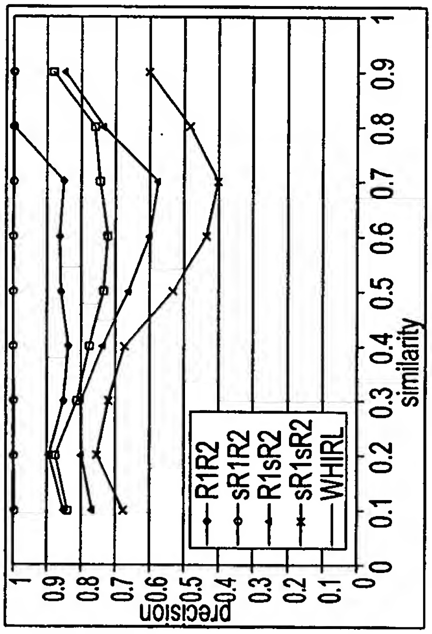


FIG. 9E

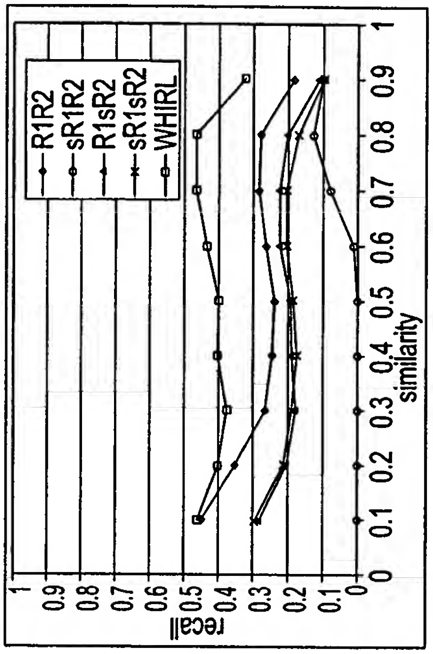


FIG. 9F

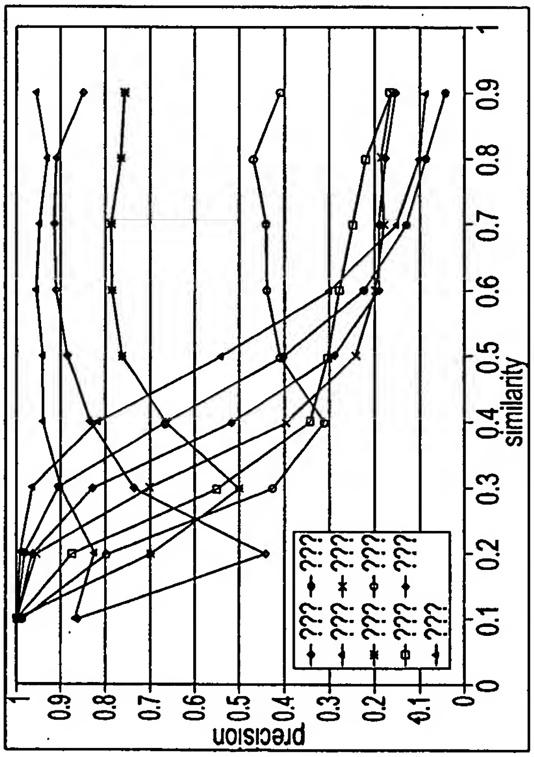


FIG. 10A

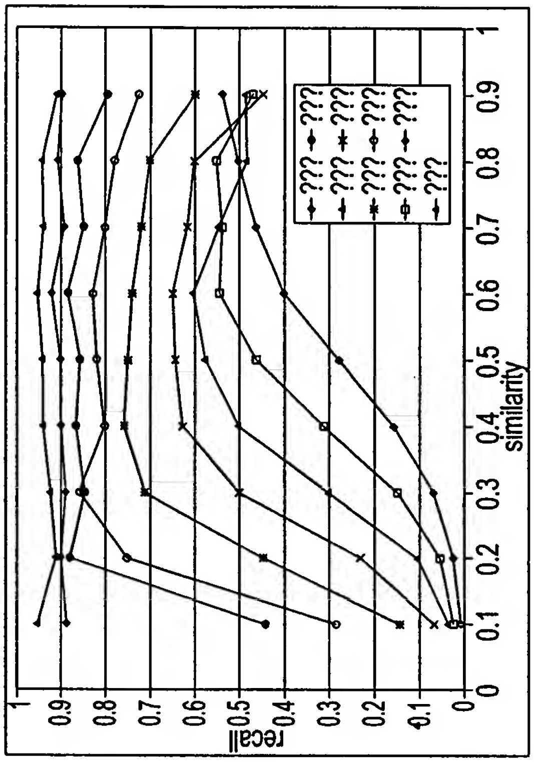


FIG. 10B

(a) Words

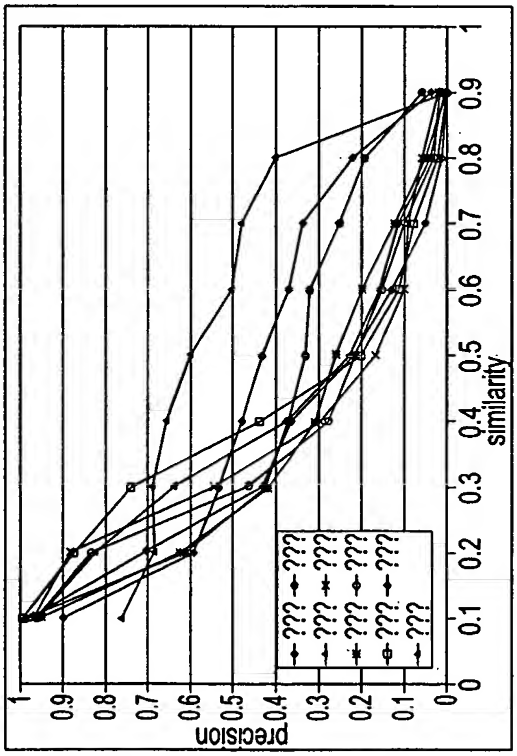


FIG. 10C

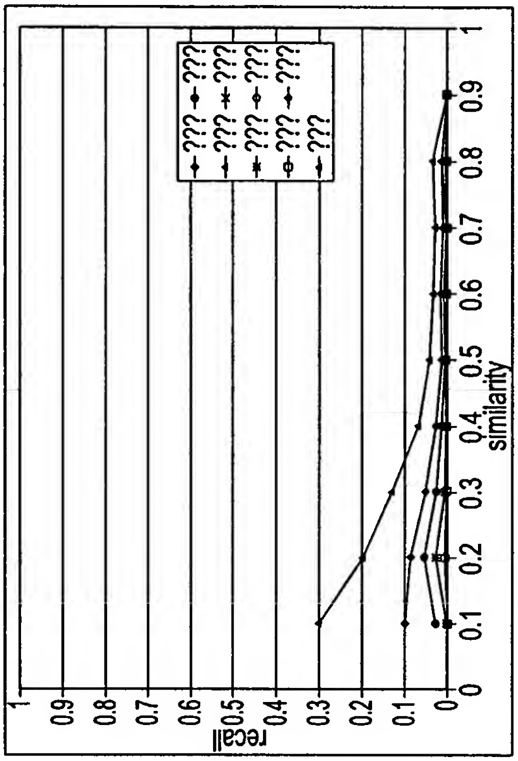


FIG. 10D

(b) Q-grams with q = 2

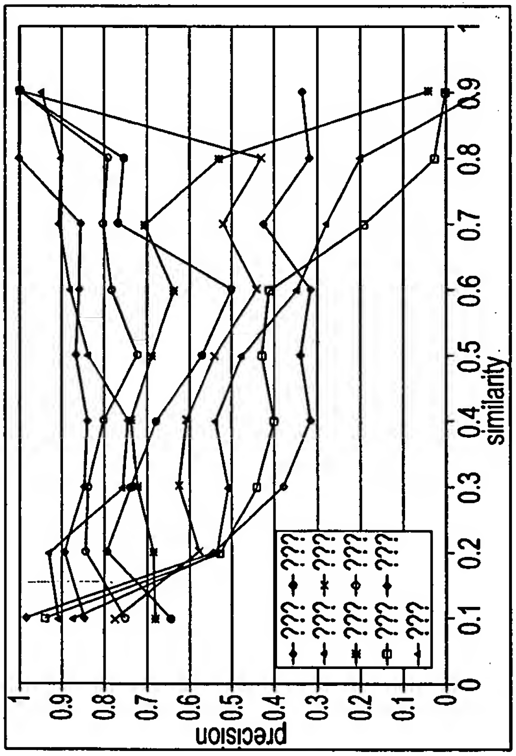


FIG. 10E

(c) Q-grams with q = 3

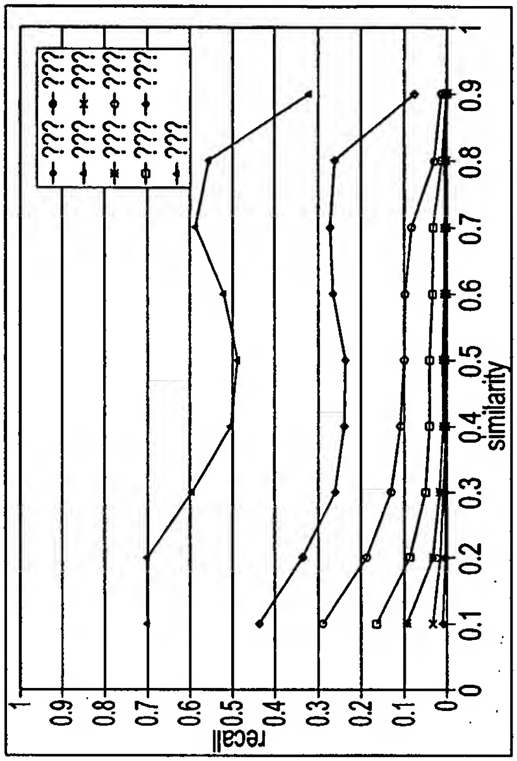


FIG. 10F

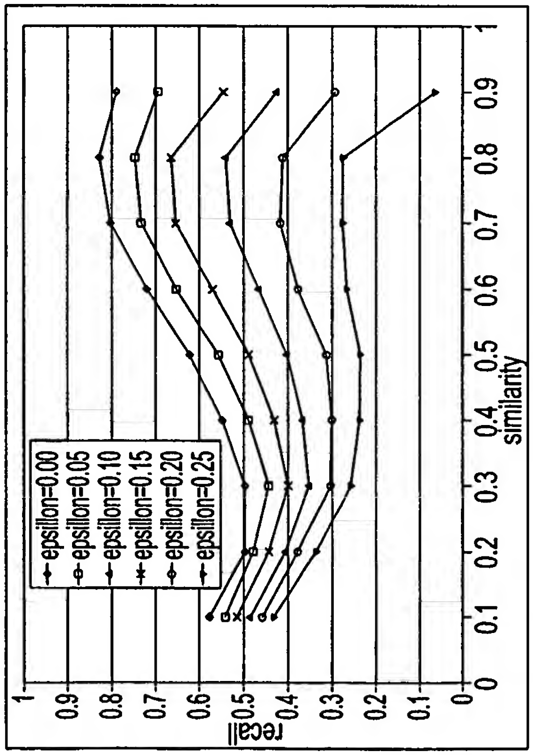


FIG. 11A

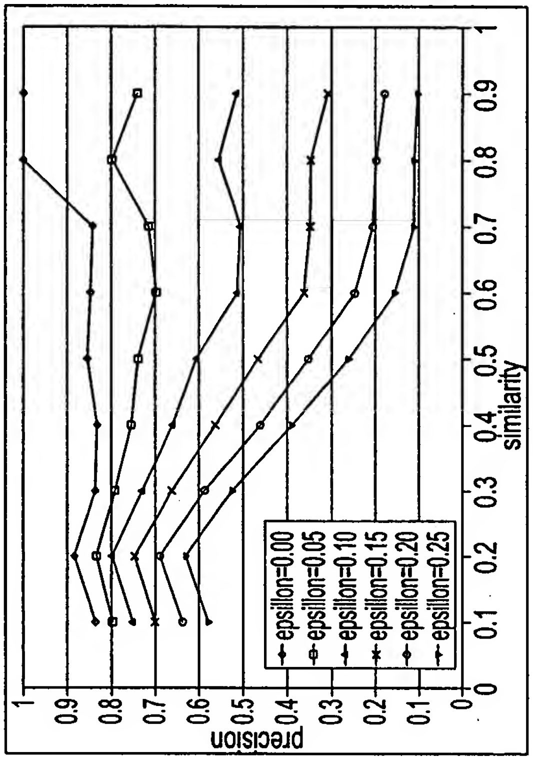
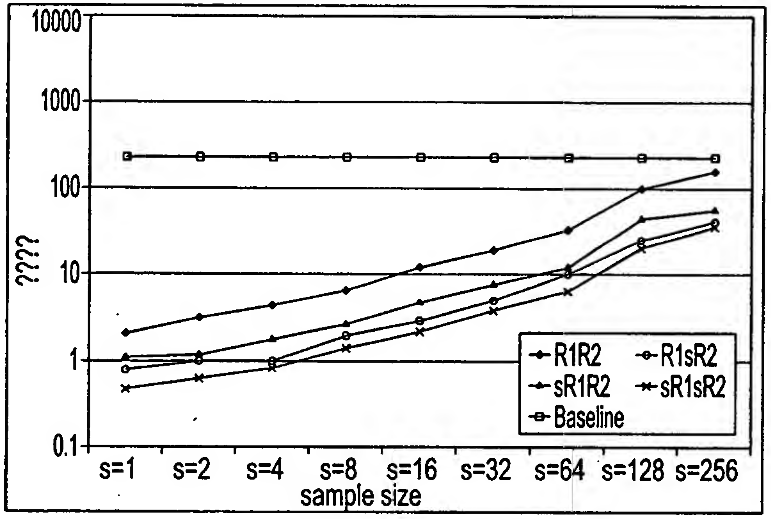
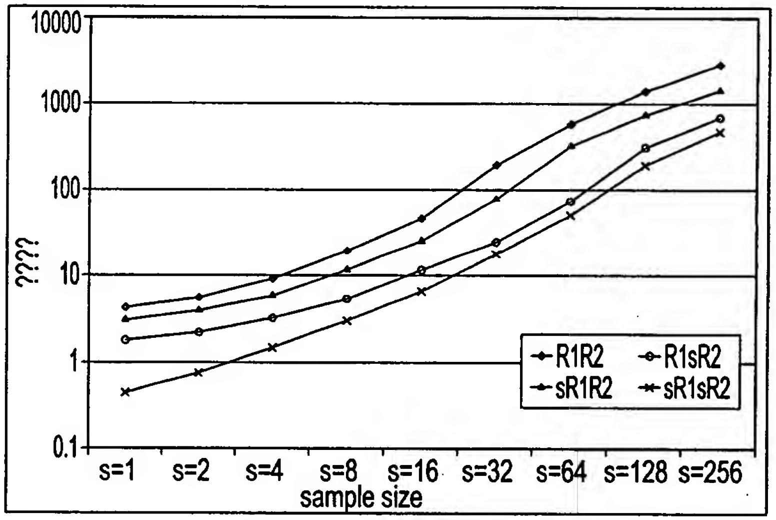


FIG. 11B



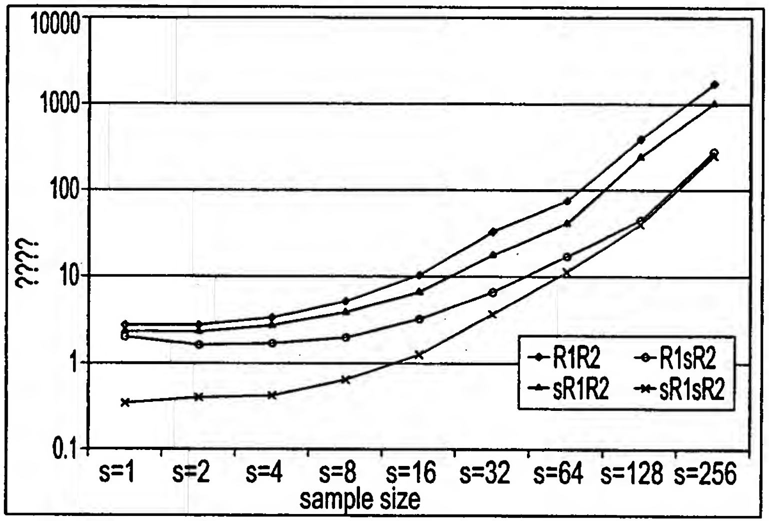
(a) Words

FIG. 12A



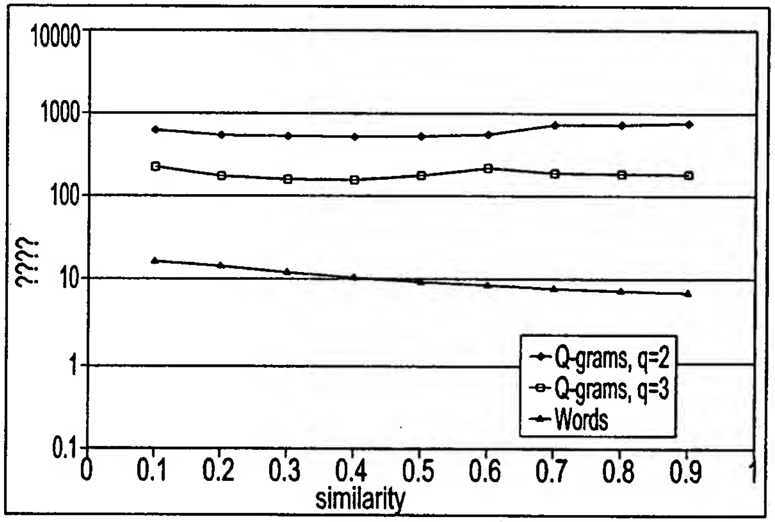
(b) Q-grams with q = 3

FIG. 12B



(c) Q-grams with q = 2

FIG. 12C



(d) WHIRL

FIG. 12D